



Busca Local e Hill Climbing

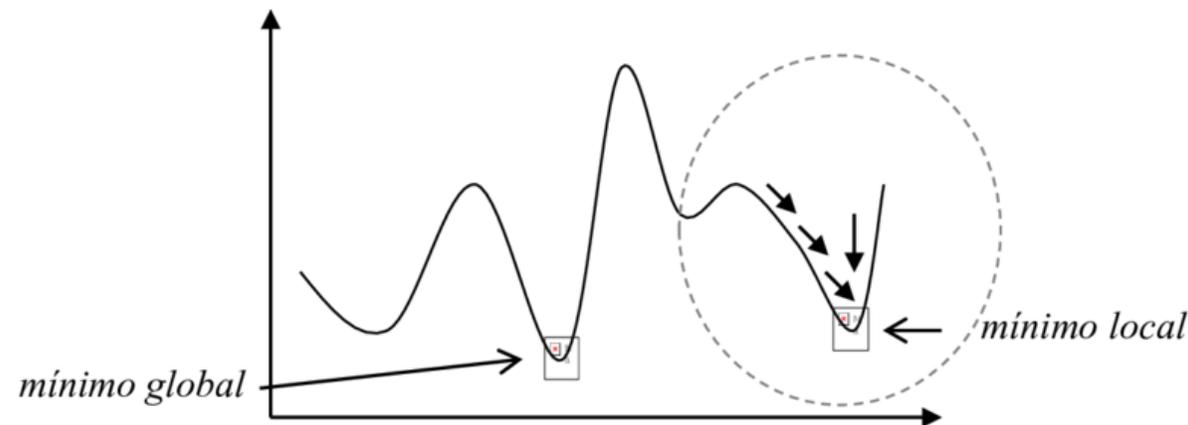
Inteligência Artificial

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Busca Local (Local Search)

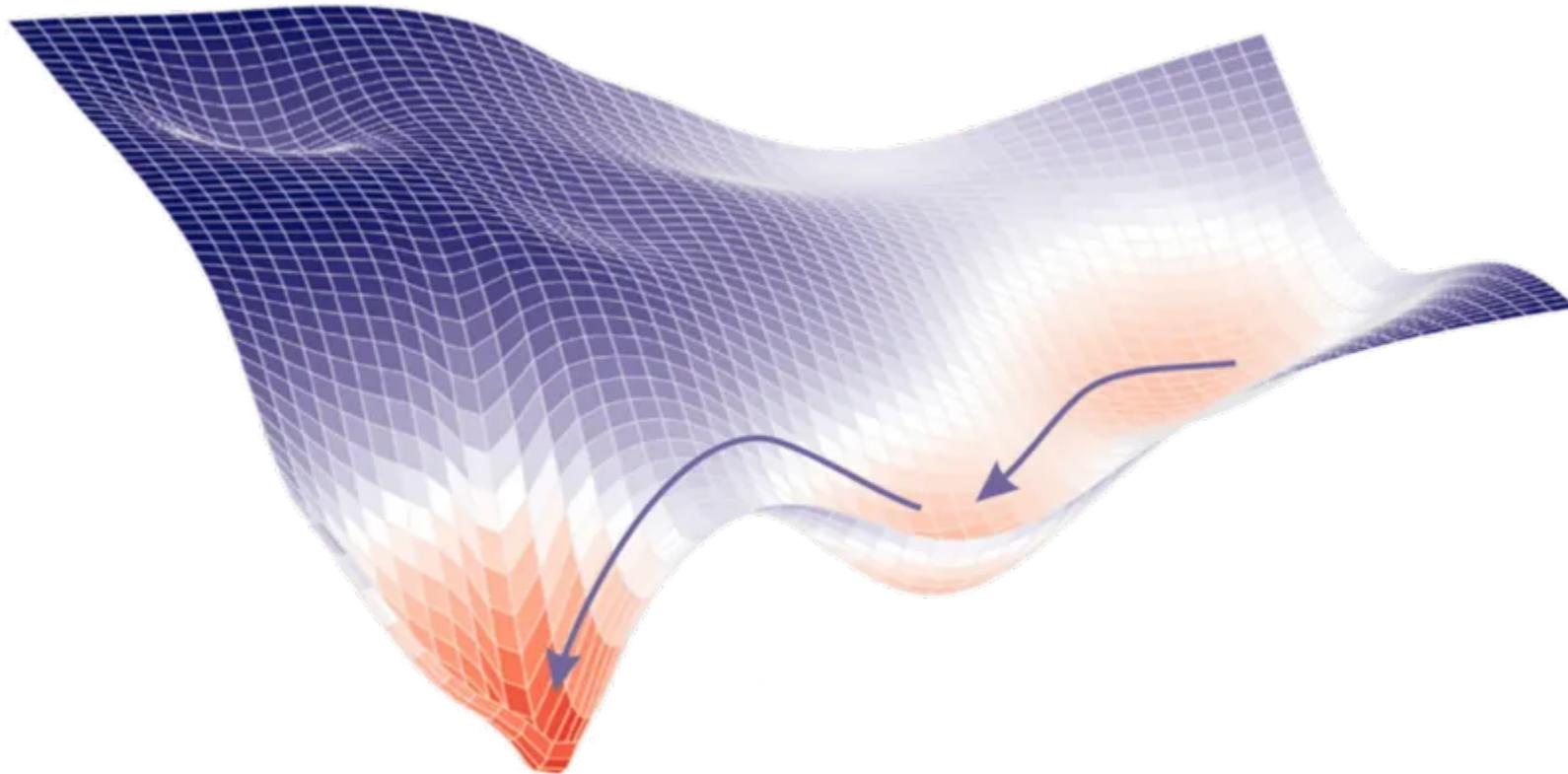
- **Objetivo:** Encontrar o **melhor estado** (solução) possível de acordo com uma função objetivo.
- A solução procurada é o **estado objetivo** (solução objetivo), sendo o **caminho** percorrido até ele **irrelevante**.
- **Mecanismo de Busca:** Melhoria incremental de uma solução candidata.
 - O algoritmo se move iterativamente, geralmente, apenas para os **estados vizinhos** desse estado atual.
 - Normalmente, o percurso não é armazenado.
- Podem encontrar soluções razoáveis em **grandes espaços de estados (contínuos)**, onde os algoritmos sistemáticos são inadequados.



Algoritmo Hill Climbing

- O Hill Climbing (ou Subida da Encosta) é um **algoritmo de busca heurística** que pertence à família dos métodos de busca local, utilizado para encontrar soluções ótimas ou quase ótimas.
 - **Busca Gulosa Local:** A cada passo, ele escolhe o movimento que otimiza (melhora) imediatamente a função objetivo, sem considerar as consequências globais.
- **Funcionamento Básico:** O algoritmo se move continuamente "para cima da montanha" (em direção ao valor crescente).
 - **Maximização (Função Objetivo):** Busca o pico mais alto (máximo global).
 - **Minimização (Função Custo):** Busca o vale mais baixo (mínimo global).
- **Finalização:** O algoritmo termina quando atinge um **pico** (ótimo local ou global), onde **não existe vizinho** com um valor mais alto na função objetivo.

Exemplo de Superfície de Otimização



Superfície (landscape) da Função Objetivo. Fonte: [primo.ai](https://www.primo.ai)

Topologia do Espaço de Estados

- **Eixo X:** Representa o **Espaço de Estados** (todas as configurações possíveis).
- **Eixo Y:** Representa o **Valor da Função Objetivo**.
- **Máximo Global (Global Maximum):** A melhor solução possível, onde a função objetivo atinge seu valor mais alto.
- **Máximo Local (Local Maximum):** Um estado que é melhor que todos os seus vizinhos, mas não é o máximo global.
- **Platô (Plateau/Flat Local Maximum):** Uma região plana onde todos os vizinhos têm o mesmo valor. Se não houver uma saída para cima, o algoritmo pode ficar paralisado.
- **Cordilheira (Ridge):** Uma sequência de máximos locais que formam uma crista estreita e ascendente, mas que é de difícil navegação para algoritmos gulosos.
- **Ombro (Shoulder):** Um platô com uma borda ascendente, que pode levar a soluções melhores.

Definição Formal

- Seja S conjunto de todas as soluções possíveis.
- Função objetivo $f : S \rightarrow \mathbb{R}$.
- **Hill Climbing**: algoritmo iterativo que, partindo de uma solução inicial s_0 , gera sucessores em sua vizinhança $N(s)$ e move-se para o melhor sucessor se ele for melhor do que a solução atual.

Pseudocódigo do Hill Climbing

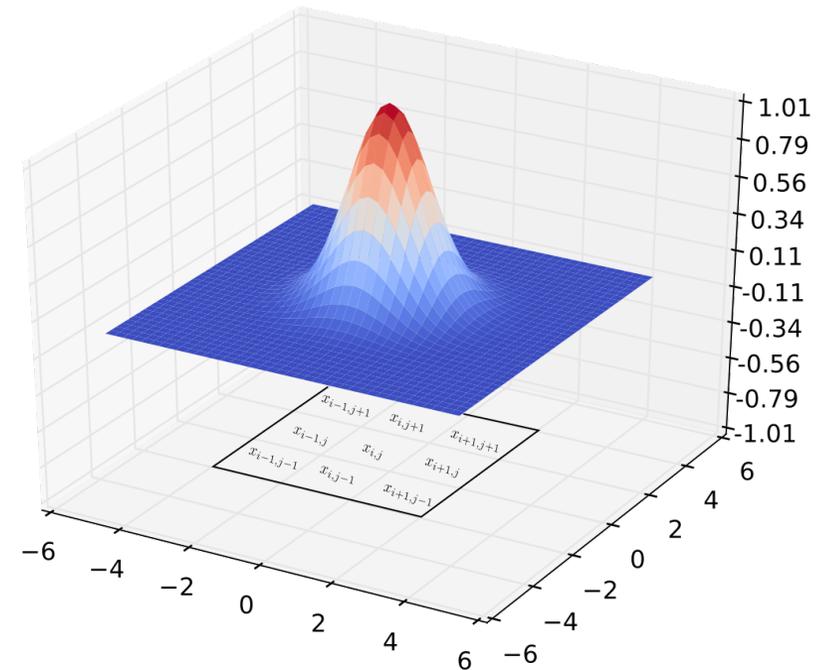
A versão mais comum é o Hill Climbing **Simples** (ou **First-Choice** Hill Climbing), que seleciona o primeiro vizinho que oferece melhoria.

```
função HILL_CLIMBING(h, s0):  
    s ← s0  
    Para v em N(s):  
        Se h(v) > h(s) então:  
            s ← v  
    retorna s
```

- **N(s)**: conjunto de vizinhos (por exemplo, troca de um elemento, incremento/decremento de coordenadas).
- Critério “>” pode ser adaptado a minimização.

Exemplo de Vizinhança

$x_{i-1,j+1}$	$x_{i,j+1}$	$x_{i+1,j+1}$
$x_{i-1,j}$	$x_{i,j}$	$x_{i+1,j}$
$x_{i-1,j-1}$	$x_{i,j-1}$	$x_{i+1,j-1}$

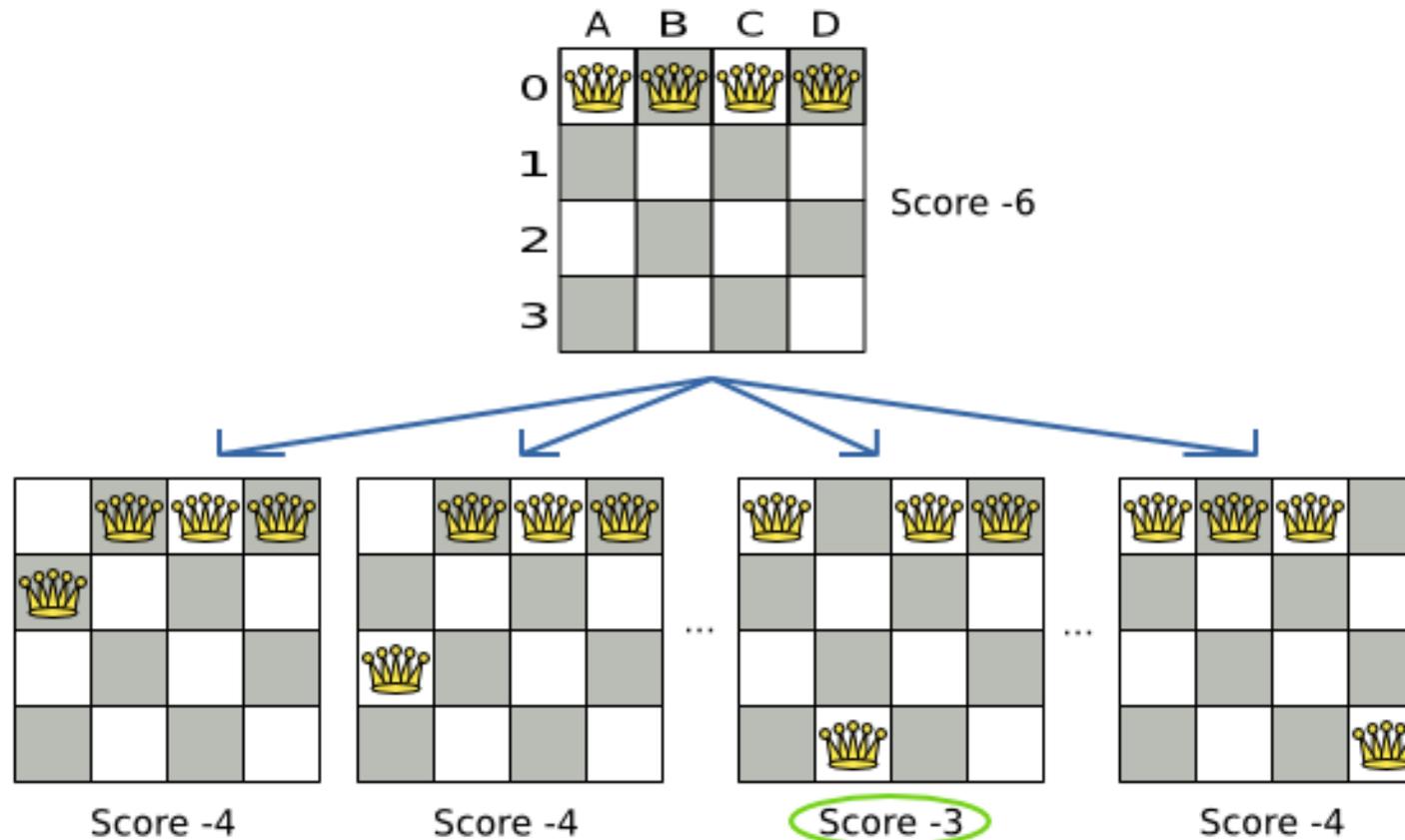


Exemplo de coordenadas para vizinhança em Hill Climbing. Fonte: [Andres Mendez-Vazquez](#)

N-Rainhas com Hill Climbing

- **Representação do Estado:** `state[i]=j`. Um vetor de N dígitos. Existe uma rainha na coluna `i`, linha `j`.
- **Função de Custo (h):** O número de pares de rainhas que se atacam.
 - O objetivo é **Minimizar h para 0**.
- **Movimento/Sucessor:** Mover uma única rainha para outro quadrado **na mesma coluna**.

Exemplo N-Rainhas



N-Rainhas com Hill Climbing. Fonte: [Optaplanner](#).

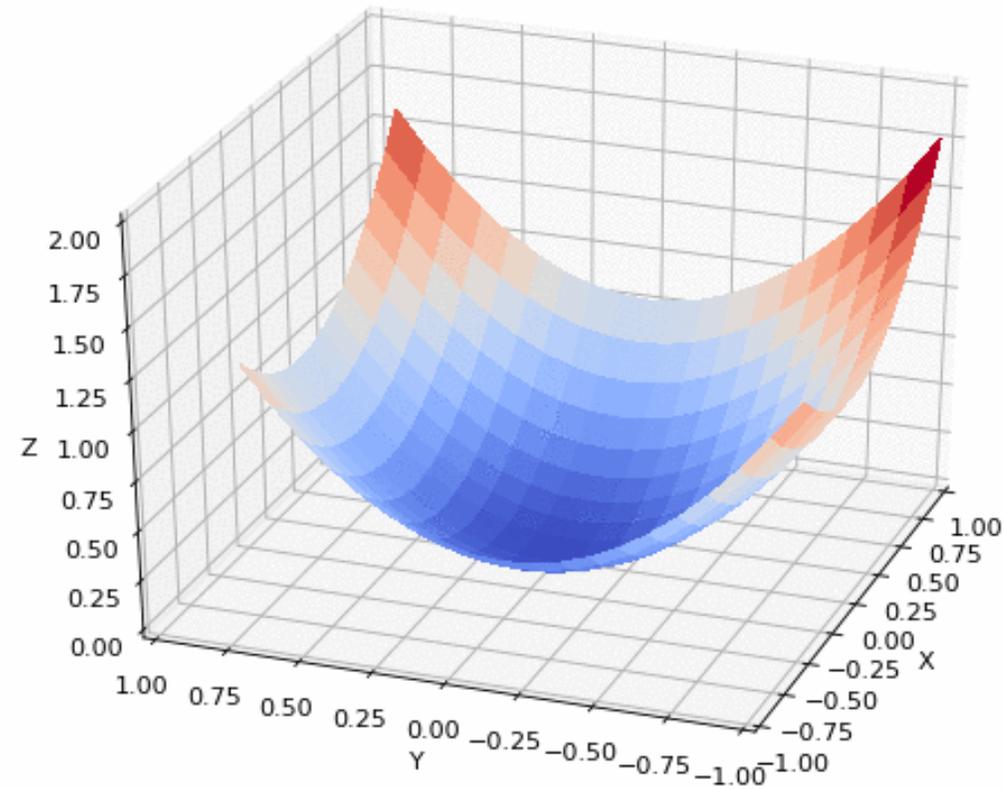
Vantagens

- **Simplicidade:** Algoritmo simples, intuitivo e de **fácil implementação**.
- **Eficiência de Memória:** Usa pouca memória (usualmente constante), pois não salva caminhos.
- **Rapidez de Convergência:** Altamente eficiente na busca por um **ótimo local** (uma solução razoável).
- **Adaptabilidade:** Pode ser aplicado a grandes espaços de busca e a problemas com **funções objetivo desconhecidas** ("caixa preta"), onde apenas a avaliação da qualidade é possível.

Aplicações

- **Otimização Combinatória:** Resolução de problemas como roteamento e alocação de recursos.
- **Inteligência Artificial em Jogos (Game AI):** Avaliação e melhoria de posições em jogos (como xadrez), buscando o próximo movimento que maximiza a vantagem.
- **Pathfinding e Robótica:** Encontrar caminhos ótimos ou mais curtos para navegação.
- **Machine Learning:** Utilizado em processos de **otimização de hiperparâmetros** para modelos de ML. Algoritmo **gradiente descendente**.
- **Projeto de Circuitos Integrados** e escalonamento de horários.

Gradiente Descendente



Exemplo de Gradiente Descendente. Fonte: [Digital Ocean](#).

Limitações e Armadilhas

A principal fraqueza do Hill Climbing é sua natureza **gulosa**, que o impede de explorar o espaço de busca de forma global, resultando na **paralisação**.

Limitação	Descrição
Máximo Local (Local Maximum)	O algoritmo fica preso em um pico que é inferior ao máximo global, pois todos os movimentos vizinhos resultam em piora.
Platô (Plateau)	Região plana onde todos os vizinhos têm o mesmo valor objetivo . O algoritmo não consegue decidir qual direção seguir e pode ser incapaz de achar a saída (exceto em casos de shoulders).
Cordilheira (Ridge)	Uma sequência de máximos locais que forma uma crista ascendente . Exige movimentos combinados que o Hill Climbing pode não ser capaz de realizar em um único passo, levando-o a parar.

Variações do Hill Climbing

As variações buscam introduzir mecanismos de **Exploração** para escapar dos máximos locais, equilibrando-a com a **Exploitação** inerente ao Hill Climbing.

- Steepest Ascent Hill-Climbing (Subida Mais Íngreme)
 - **Mecanismo:** Em cada iteração, avalia **todos** os vizinhos gerados e seleciona aquele que oferece a **melhor melhoria** (maior valor objetivo ou menor custo).
- Stochastic Hill Climbing (Subida Estocástica)
 - **Mecanismo:** Seleciona um vizinho para se mover de **forma aleatória** entre o subconjunto de vizinhos que são **melhores** que o estado atual.
- Hill-Climbing with Random Restarts (Com Reinício Aleatório)
 - **Mecanismo:** Executa várias instâncias do Hill Climbing, cada uma iniciando a partir de um **estado inicial diferente e aleatório**.

Conclusão

- O Hill Climbing (Subida da Encosta) é um **algoritmo de busca local de estado único** focado em **Problemas de Otimização**, onde o caminho até a solução é irrelevante.
- **Simplicidade e Facilidade** de Implementação.
- **Eficiência de Memória**: Utiliza **pouca memória** (usualmente constante), sendo adequado para grandes ou infinitos espaços de estados.

Limitação	Explicação
Padrões locais	Pode ficar preso em ótimos que não são globais.
Sensibilidade à inicialização	Resultado depende da solução inicial s_0 .
Vizinhança limitada	Se a vizinhança for pequena, melhorias podem ser insuficientes.
Convergência lenta	Em espaços de alta dimensionalidade pode exigir muitas iterações.